

1. ให้นักศึกษานำชุดข้อมูล Nutrition จากแฟ้มข้อมูล เข้าสู่โปรแกรม และแสดงภาพหน้าจอ ของโปรแกรม นักศึกษาที่ใช้ทาแบบฝึกปฏิบัติการ ตามข้อ 1-14 (สิ่งที่ต้องส่งเป็นการบ้าน) *** โดยกำหนดค่า attributes ที่ท ำ ตามข้อ 1-14 ดังนี้

```
[3] import pandas as pd
import matplotlib.pyplot as plt

nutri = pd.read_csv("nutriRaw.csv")

[5] nutri.head()
```

	Unnamed: 0	gender	situation	tea	coffee	height	weight	age	meat	fish	raw_fruit	cooked_fruit_veg	chocol	fat
0	0	2	1	0	0	151	58	72	4	3	1	4	5	6
1	1	2	1	1	1	162	60	68	5	2	5	5	1	4
2	2	2	1	0	4	162	75	78	3	1	5	2	5	4
3	3	2	1	0	0	154	45	91	0	4	4	0	3	2
4	4	2	1	2	1	154	50	65	5	3	5	5	3	2

Next steps: [Generate code with nutri](#) [View recommended plots](#) [New interactive sheet](#)

```
[6] nutri.drop(axis=1, columns='Unnamed: 0', inplace=True)

nutri.head(5)
```

	gender	situation	tea	coffee	height	weight	age	meat	fish	raw_fruit	cooked_fruit_veg	chocol	fat
0	2	1	0	0	151	58	72	4	3	1	4	5	6
1	2	1	1	1	162	60	68	5	2	5	5	1	4
2	2	1	0	4	162	75	78	3	1	5	2	5	4
3	2	1	0	0	154	45	91	0	4	4	0	3	2
4	2	1	2	1	154	50	65	5	3	5	5	3	2

Next steps: [Generate code with nutri](#) [View recommended plots](#) [New interactive sheet](#)

```
[8] nutri.shape
```

```
→ (226, 13)
```

```
[9] nutri.info()
```

```
→ <class 'pandas.core.frame.DataFrame'>  
RangeIndex: 226 entries, 0 to 225  
Data columns (total 13 columns):  
#   Column                Non-Null Count  Dtype  
---  ---  
0   gender                 226 non-null   int64  
1   situation              226 non-null   int64  
2   tea                    226 non-null   int64  
3   coffee                 226 non-null   int64  
4   height                 226 non-null   int64  
5   weight                 226 non-null   int64  
6   age                    226 non-null   int64  
7   meat                   226 non-null   int64  
8   fish                   226 non-null   int64  
9   raw_fruit              226 non-null   int64  
10  cooked_fruit_veg       226 non-null   int64  
11  chocol                  226 non-null   int64  
12  fat                     226 non-null   int64  
dtypes: int64(13)  
memory usage: 23.1 KB
```

```
[10] dictGender = {1: 'Male', 2: 'Female'}
```

```
[11] nutri['gender'] = nutri['gender'].replace(dictGender).astype('category')  
  
nutri.head(1)
```



	gender	situation	tea	coffee	height	weight	age	meat	fish	raw_fruit	cooked_fruit_veg	chocol	fat	
0	Female		1	0	0	151	58	72	4	3	1	4	5	6

Next steps:

[Generate code with nutri](#)[View recommended plots](#)[New interactive sheet](#)

```
nutri.info()
```



```
<class 'pandas.core.frame.DataFrame'>  
RangeIndex: 226 entries, 0 to 225  
Data columns (total 13 columns):  
#   Column          Non-Null Count  Dtype  
---  ---  
0   gender           226 non-null   category  
1   situation         226 non-null   int64  
2   tea               226 non-null   int64  
3   coffee            226 non-null   int64  
4   height            226 non-null   int64  
5   weight            226 non-null   int64  
6   age               226 non-null   int64  
7   meat              226 non-null   int64  
8   fish              226 non-null   int64  
9   raw_fruit         226 non-null   int64  
10  cooked_fruit_veg  226 non-null   int64  
11  chocol            226 non-null   int64  
12  fat               226 non-null   int64  
dtypes: category(1), int64(12)  
memory usage: 21.7 KB
```

```
unique_counts = pd.DataFrame.from_records([(col, nutri[col].nunique()) for col in nutri.columns],  
columns=['Column_Name', 'Unique_Count']).sort_values(by=['Unique_Count'])
```


unique_counts

	Column_Name	Unique_Count
0	gender	2
1	situation	3
3	coffee	6
7	meat	6
8	fish	6
9	raw_fruit	6
10	cooked_fruit_veg	6
11	chocol	6
12	fat	8
2	tea	9
6	age	27
4	height	37
5	weight	50



```
[15] nutri['BinningSituation'] = pd.cut(nutri['situation'], bins=4)

[16] nutri['BinningSituation']
```

 **BinningSituation**


0	(0.998, 1.5]
1	(0.998, 1.5]
2	(0.998, 1.5]
3	(0.998, 1.5]
4	(0.998, 1.5]
...	...
221	(0.998, 1.5]
222	(1.5, 2.0]
223	(1.5, 2.0]
224	(0.998, 1.5]
225	(1.5, 2.0]

226 rows × 1 columns

dtype: category

```
[17] nutri['BinningSituation_1'] = pd.cut(nutri['situation'], bins=4, labels=['Single', 'Living with spouse', 'Living with family', 'Living with someone else'])

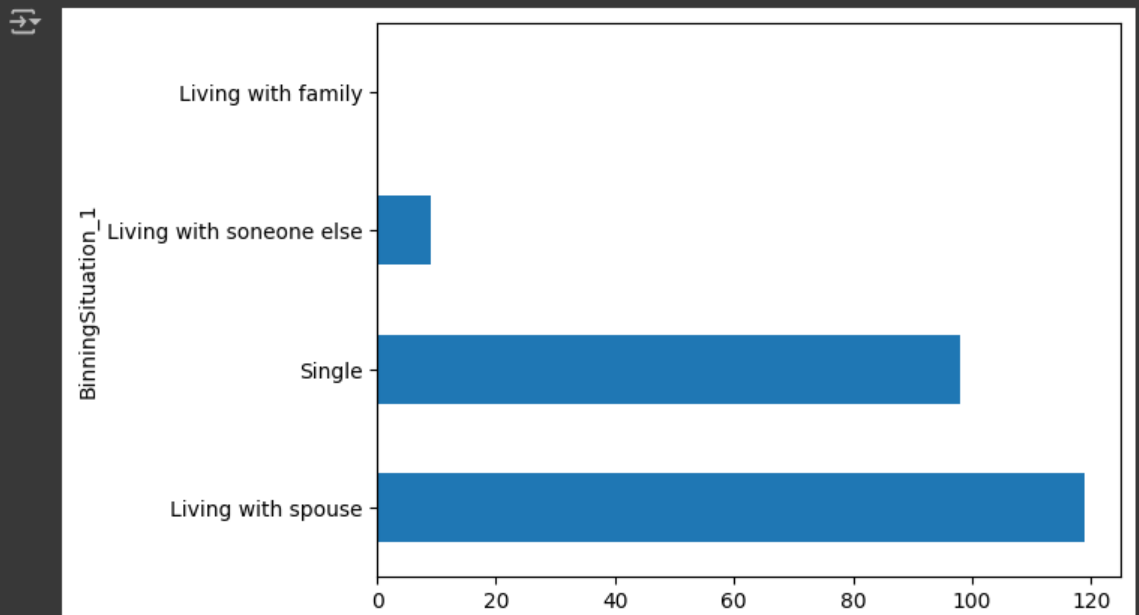
nutri.head(10)
```

 **gender situation tea coffee height weight age meat fish raw_fruit cooked_fruit_veg chocol fat BinningSituation BinningSituation_1**

0	Female	1	0	0	151	58	72	4	3	1	4	5	6	(0.998, 1.5]	Single
1	Female	1	1	1	162	60	68	5	2	5	5	1	4	(0.998, 1.5]	Single
2	Female	1	0	4	162	75	78	3	1	5	2	5	4	(0.998, 1.5]	Single
3	Female	1	0	0	154	45	91	0	4	4	0	3	2	(0.998, 1.5]	Single
4	Female	1	2	1	154	50	65	5	3	5	5	3	2	(0.998, 1.5]	Single
5	Female	1	2	0	159	66	82	4	2	5	5	1	3	(0.998, 1.5]	Single
6	Female	1	2	0	160	66	74	3	3	5	5	5	6	(0.998, 1.5]	Single
7	Female	1	0	2	163	66	73	4	2	5	5	1	6	(0.998, 1.5]	Single
8	Female	1	0	3	154	60	89	4	3	5	5	5	6	(0.998, 1.5]	Single
9	Female	1	0	2	160	77	87	2	3	5	4	0	3	(0.998, 1.5]	Single

Next steps: [Generate code with nutri](#) [View recommended plots](#) [New interactive sheet](#)

```
nutri['BinningSituation_1'].value_counts().plot(kind='barh')
plt.show()
```






```
nutri['BinningMeat'] = pd.cut(nutri['meat'], bins=6, labels=['Never', 'Less than once a week', 'Once a week', '2-3 times a week', '4-6 times a week', 'Every day'])
nutri['BinningFish'] = pd.cut(nutri['fish'], bins=6, labels=['Never', 'Less than once a week', 'Once a week', '2-3 times a week', '4-6 times a week', 'Every day'])
nutri['BinningRawFruit'] = pd.cut(nutri['raw_fruit'], bins=6, labels=['Never', 'Less than once a week', 'Once a week', '2-3 times a week', '4-6 times a week', 'Every day'])
nutri['BinningCookedFruitVeg'] = pd.cut(nutri['cooked_fruit_veg'], bins=6, labels=['Never', 'Less than once a week', 'Once a week', '2-3 times a week', '4-6 times a week', 'Every day'])
nutri['BinningChocol'] = pd.cut(nutri['chocol'], bins=6, labels=['Never', 'Less than once a week', 'Once a week', '2-3 times a week', '4-6 times a week', 'Every day'])
nutri['BinningFat'] = pd.cut(nutri['fat'], bins=6, labels=['Butter', 'Margarine', 'Peanut oil', 'Sunflower oil', 'Olive oil', 'Mix of vegetable', 'Coza oil', 'Duck or goose fat'])
```

```
nutri.head(10)
```

	gender	situation	tea	coffee	height	weight	age	meat	fish	raw_fruit	...	chocol	fat	BinningSituation	BinningSituation_1	BinningMeat	BinningFish	BinningRawFruit	BinningCookedFruitVeg	BinningChocol	BinningFat
0	Female	1	0	0	151	58	72	4	3	1	...	5	6	(0.998, 1.5]	Single	4-6 times a week	2-3 times a week	Less than once a week	4-6 times a week	Every day	Mix of vegetable
1	Female	1	1	1	162	60	68	5	2	5	...	1	4	(0.998, 1.5]	Single	Every day	Once a week	Every day	Every day	Less than once a week	Sunflower oil
2	Female	1	0	4	162	75	78	3	1	5	...	5	4	(0.998, 1.5]	Single	2-3 times a week	Less than once a week	Every day	Once a week	Every day	Sunflower oil
3	Female	1	0	0	154	45	91	0	4	4	...	3	2	(0.998, 1.5]	Single	Never	4-6 times a week	4-6 times a week	Never	2-3 times a week	Margarine
4	Female	1	2	1	154	50	65	5	3	5	...	3	2	(0.998, 1.5]	Single	Every day	2-3 times a week	Every day	Every day	2-3 times a week	Margarine
5	Female	1	2	0	159	66	82	4	2	5	...	1	3	(0.998, 1.5]	Single	4-6 times a week	Once a week	Every day	Every day	Less than once a week	Peanut oil
6	Female	1	2	0	160	66	74	3	3	5	...	5	6	(0.998, 1.5]	Single	2-3 times a week	2-3 times a week	Every day	Every day	Every day	Mix of vegetable
7	Female	1	0	2	163	66	73	4	2	5	...	1	6	(0.998, 1.5]	Single	4-6 times a week	Once a week	Every day	Every day	Less than once a week	Mix of vegetable
8	Female	1	0	3	154	60	89	4	3	5	...	5	6	(0.998, 1.5]	Single	4-6 times a week	2-3 times a week	Every day	Every day	Every day	Mix of vegetable
9	Female	1	0	2	160	77	87	2	3	5	...	0	3	(0.998, 1.5]	Single	Once a week	2-3 times a week	Every day	4-6 times a week	Never	Peanut oil




10 rows × 21 columns

```
[22] pd.crosstab(nutri.gender, nutri.situation)
```




situation	1	2	3
gender			
Female	78	56	7
Male	20	63	2

```
[23] pd.crosstab(nutri.gender, nutri.situation, margins=True)
```




situation	1	2	3	All
gender				
Female	78	56	7	141
Male	20	63	2	85
All	98	119	9	226

```
[24] nutri['coffee'].mean()
```



```
1.6194690265486726
```

```
[25] nutri['coffee'].quantile(q=[0.25, 0.5, 0.75])
```



	coffee
0.25	1.0
0.50	2.0
0.75	2.0

dtype: float64

```
[26] nutri['coffee'].var()
```

```
↔ 1.5701081612586039
```

```
[27] round(nutri['coffee'].var(), 2)
```

```
↔ 1.57
```

```
[28] nutri['coffee'].std()
```

```
↔ 1.2530395689117737
```

```
[29] round(nutri['coffee'].std(), 2)
```

```
↔ 1.25
```

```
[30] nutri.to_csv('nutriOutput.csv')
```


2. ให้นักศึกษาระบุชนิดข้อมูล attributes ของ Nutrition จากตารางข้างต้น ว่าเป็นประเภทข้อมูลแบบไหน (Qualitative, Discrete quantitative, Continuous quantitative) พร้อมอธิบายเหตุผลที่กำหนด

gender situation tea coffee height weight age meat fish raw_fruit cooked_fruit_veg chocol fat

Gender: Qualitative เพศเป็นประเภทข้อมูลที่แสดงถึงหมวดหมู่ (เช่น ชาย หญิง) ซึ่งไม่สามารถวัดเป็นตัวเลขได้

Situation: Qualitative สถานการณ์เป็นคำอธิบายเชิงหมวดหมู่ที่ไม่สามารถหาค่าทางตัวเลขได้

Tea: Discrete Quantitative จำนวนถ้วยหรือปริมาณการดื่มชา สามารถนับได้ ทำให้เป็นข้อมูลเชิงปริมาณแบบไม่ต่อเนื่อง

Coffee: Discrete Quantitative การดื่มกาแฟสามารถนับเป็นจำนวนถ้วยหรือจำนวนครั้งได้

Height: Continuous Quantitative ความสูงสามารถมีค่าภายในช่วงตัวเลขและรวมถึงทศนิยมได้

Weight: Continuous Quantitative น้ำหนักสามารถวัดเป็นตัวเลขที่ละเอียดและมีค่าทศนิยมได้

Age: Discrete Quantitative อายุโดยปกติจะนับเป็นจำนวนปี ซึ่งเป็นข้อมูลเชิงปริมาณแบบไม่ต่อเนื่อง

Meat: Qualitative การบริโภคเนื้อสัตว์เป็นหมวดหมู่

Fish: Qualitative การบริโภคปลาจัดเป็นหมวดหมู่

Raw Fruit: Qualitative เป็นการแบ่งหมวดหมู่ของการบริโภคอาหาร ซึ่งเป็นข้อมูลเชิงคุณภาพ

Cooked Fruit Veg: Qualitative เป็นหมวดหมู่ของการบริโภคอาหารที่จัดอยู่ในข้อมูลเชิงคุณภาพ

Chocolate: Qualitative การบริโภคช็อกโกแลตเป็นหมวดหมู่เชิงคุณภาพ

Fat: Qualitative ประเภทหรือการมีไขมันในอาหารจัดเป็นข้อมูลเชิงหมวดหมู่